

```

!pip install -q kagglehub
import kagglehub
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.impute import SimpleImputer
from sklearn.ensemble import RandomForestRegressor
from sklearn.feature_selection import SelectKBest, f_regression
from sklearn.metrics import mean_squared_error

path = kagglehub.dataset_download("zkskhurram/fifa-and-football-complete-dataset-19302022")
print("Path to dataset files:", path)

import os
print("Files in dataset:", os.listdir(path))

```

Downloading to /root/.cache/kagglehub/datasets/zkskhurram/fifa-and-football-complete-dataset-19302022/2.archive...
 100%[██████████] 4.20k/4.20k [00:00<00:00, 5.26MB/s]Extracting files...
 Path to dataset files: /root/.cache/kagglehub/datasets/zkskhurram/fifa-and-football-complete-dataset-19302022/versions/2
 Files in dataset: ['fifa_world_rankings_jan_2026.csv', 'fifa_world_cup_history.csv', 'football_major_competitions.csv', 'fif

```

import os
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.impute import SimpleImputer
from sklearn.ensemble import RandomForestRegressor
from sklearn.feature_selection import SelectKBest, f_regression, RFE
from sklearn.metrics import mean_squared_error

files = os.listdir(path)
print("Available files:", files)

csv_file = [f for f in files if f.endswith('.csv')][0]
df = pd.read_csv(os.path.join(path, csv_file), low_memory=False)

print("Data Shape:", df.shape)
display(df.head(3))

numeric_cols = df.select_dtypes(include=[np.number]).columns.tolist()

if len(numeric_cols) > 1:
    target_col = numeric_cols[-1]
    feature_cols = numeric_cols[:-1]
    X = df[feature_cols].copy()
    y = df[target_col].copy()

    imputer = SimpleImputer(strategy='mean')
    X_imputed = pd.DataFrame(imputer.fit_transform(X), columns=feature_cols)

    valid_indices = ~y.isna()
    X_imputed = X_imputed[valid_indices]
    y = y[valid_indices]

    X_train, X_test, y_train, y_test = train_test_split(X_imputed, y, test_size=0.2, random_state=42)

    print("\n--- Step 3: GridSearchCV Training ---")
    param_grid = {
        'n_estimators': [50, 100],
        'max_depth': [5, 10]
    }
    rf = RandomForestRegressor(random_state=42)
    grid_search = GridSearchCV(estimator=rf, param_grid=param_grid, cv=3, scoring='neg_mean_squared_error')
    grid_search.fit(X_train, y_train)

    best_model = grid_search.best_estimator_
    print("Best Parameters:", grid_search.best_params_)
    y_pred_base = best_model.predict(X_test)
    print("Base Model MSE:", mean_squared_error(y_test, y_pred_base))

    print("\n--- Step 4: Feature Selection ---")

    k_features = min(5, len(feature_cols))

```

```

selector_kbest = SelectKBest(score_func=f_regression, k=k_features)
X_train_kbest = selector_kbest.fit_transform(X_train, y_train)
X_test_kbest = selector_kbest.transform(X_test)
print("Selected Features by SelectKBest:", list(np.array(feature_cols)[selector_kbest.get_support()])))

rfe = RFE(estimator=RandomForestRegressor(random_state=42), n_features_to_select=k_features)
X_train_rfe = rfe.fit_transform(X_train, y_train)
X_test_rfe = rfe.transform(X_test)
print("Selected Features by RFE:", list(np.array(feature_cols)[rfe.support_]))

print("\n--- Step 5: Model Evaluation on Selected Features ---")
model_kbest = RandomForestRegressor(random_state=42)
model_kbest.fit(X_train_kbest, y_train)
print("MSE using SelectKBest features:", mean_squared_error(y_test, model_kbest.predict(X_test_kbest)))

model_rfe = RandomForestRegressor(random_state=42)
model_rfe.fit(X_train_rfe, y_train)
print("MSE using RFE features:", mean_squared_error(y_test, model_rfe.predict(X_test_rfe)))
else:
    print("Please check your numerical columns for the target variable.")

```

Available files: ['fifa_world_rankings_jan_2026.csv', 'fifa_world_cup_history.csv', 'football_major_competitions.csv', 'fifa']
 Data Shape: (30, 7)

	Rank	Country	Points	Confederation	World_Cup_Titles	Best_WC_Finish	Continental_Titles	
0	1	Spain	1877.18	UEFA	1	Champions (2010)	12	
1	2	Argentina	1873.33	CONMEBOL	3	Champions (2022)	15	
2	3	France	1870.00	UEFA	2	Champions (2018)	3	

--- Step 3: GridSearchCV Training ---

Best Parameters: {'max_depth': 10, 'n_estimators': 100}

Base Model MSE: 12.939616666666666

--- Step 4: Feature Selection ---

Selected Features by SelectKBest: [np.str_('Rank'), np.str_('Points'), np.str_('World_Cup_Titles')]

Selected Features by RFE: [np.str_('Rank'), np.str_('Points'), np.str_('World_Cup_Titles')]

--- Step 5: Model Evaluation on Selected Features ---

MSE using SelectKBest features: 12.939616666666666

MSE using RFE features: 12.939616666666666